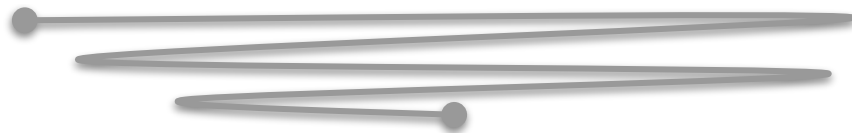


Laboratorio de Datos



Previo a SQL



2do. Cuatrimestre
2023



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

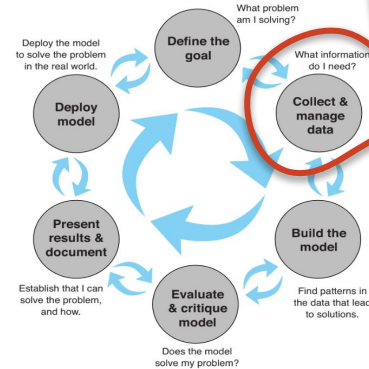


Recorrido de la materia (hasta ahora)

- ✓ Lenguaje de programación para trabajar en nuestros proyectos



- ✓ Etapas de un proyecto de Ciencias de Datos



- ✓ Modelado de Datos



- ✓ Representación de los Datos

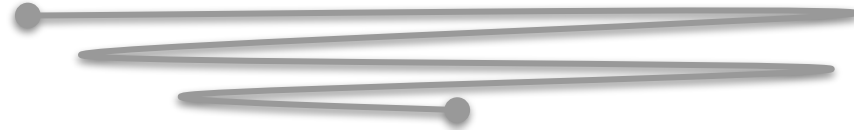
Materia

Codigo	Nombre
1	Laboratorio de Datos
2	Análisis II
3	Álgebra Lineal

Unidad

Codigo Materia	Título	Descripción
1	Administración de datos	Obtención y Manejo de los datos
1	Modelos Explicativos	Construcción de modelos explicativos
1	Modelos Predictivos	Construcción de modelos predictivos
2	Integrales sobre curvas y superficies	Integrales en múltiples variables
2	Ecuaciones Diferenciales	Ecuaciones diferenciales

Trabajo en equipo

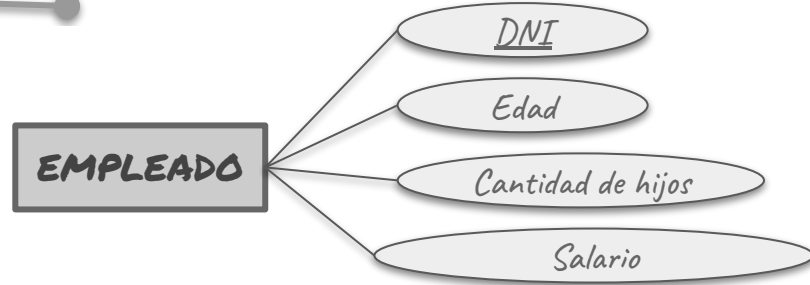


- ✓ Conformar grupos de 3 integrantes.
- ✓ Van a tener que escribir código
- ✓ Anotar las respuestas a las preguntas dentro del mismo código



Actividad Nro. 00

✓ Contamos con el siguiente DER.



✓ Implementarlo en Python. Para ello, generar una matriz (de 3 filas x 4 columnas) denominada empleado_01 con los siguientes datos (todos de tipo entero).

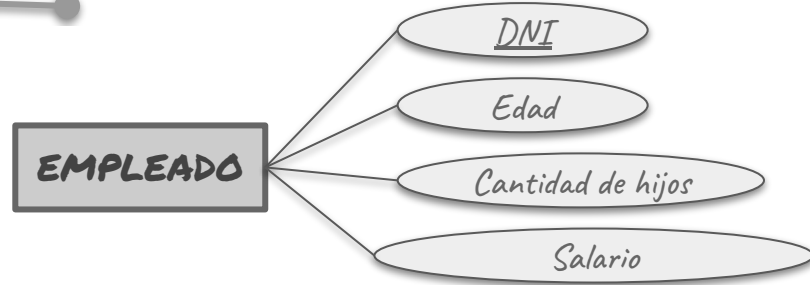
Importante. Implementar la matriz como lista de filas.

DNI	Edad	Cantidad de hijos	Salario
20.222.333	45	2	20.000
33.456.234	40	0	25.000
45.432.345	41	1	10.000

Actividad Nro. 00



✓ Contamos con el siguiente DER.



✓ Implementarlo en Python. Para ello, generar una matriz (de 3 filas x 4 columnas) denominada empleado_01 con los siguientes datos (todos de tipo entero).

Importante. Implementar la matriz como lista de filas.

<i>DNI</i>	<i>Edad</i>	<i>Cantidad de hijos</i>	<i>Salario</i>
20.222.333	45	2	20.000
33.456.234	40	0	25.000
45.432.345	41	1	10.000

Actividad Nro. 01

- ✓ Programar la función `superanSalarioActividad01` en Python. Debe tomar como entrada la matriz `empleado_01` y devolver una nueva matriz (con las 4 columnas) conteniendo a aquellos empleados que ganan un salario > \$15.000. Sólo está permitido utilizar comandos básicos de Python, es decir, ciclos (`while/for`), condicionales (`if`), etc. No está permitido importar funciones de bibliotecas.
- ✓ `superanSalarioActividad01(empleado_01)` debería devolver

20.222.333	45	2	20.000
33.456.234	40	0	25.000

- ✓ ¿Cuánto les costó implementar la función?

Actividad Nro. 01

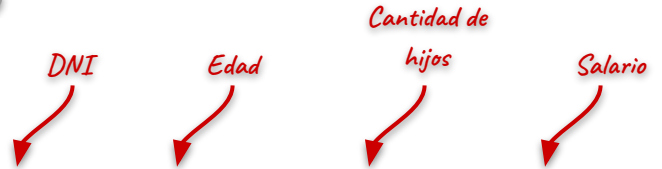


- ✓ Programar la función `superanSalarioActividad01` en Python. Debe tomar como entrada la matriz `empleado_01` y devolver una nueva matriz (con las 4 columnas) conteniendo a aquellos empleados que ganan un salario $> \$15.000$. Sólo está permitido utilizar comandos básicos de Python, es decir, ciclos (`while/for`), condicionales (`if`), etc. No está permitido importar funciones de bibliotecas.
- ✓ `superanSalarioActividad01(empleado_01)` debería devolver
- ✓ ¿Cuánto les costó implementar la función?

20.222.333	45	2	20.000
33.456.234	40	0	25.000

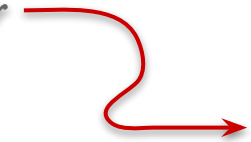
Actividad Nro. 02

- ✓ ¿Qué pasa si se agregan más filas a la matriz?
Probar con el siguiente ejemplo (denominar a la nueva matriz empleado_02).



20.222.333	45	2	20.000
33.456.234	40	0	25.000
45.432.345	41	1	10.000
43.967.304	37	0	12.000
42.236.276	36	0	18.000

- ✓ La función `superanSalarioActividad01(empleado_02)` ¿continúa funcionando? Debería devolver



20.222.333	45	2	20.000
33.456.234	40	0	25.000
42.236.276	36	0	18.000

Actividad Nro. 02



- ✓ ¿Qué pasa si se agregan más filas a la matriz?
Probar con el siguiente ejemplo (denominar a la nueva matriz empleado_02).

matriz?

minar a la

DNI

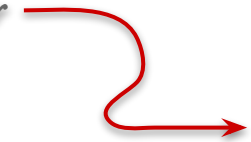
Edad

Cantidad de hijos

Salario

20.222.333	45	2	20.000
33.456.234	40	0	25.000
45.432.345	41	1	10.000
43.967.304	37	0	12.000
42.236.276	36	0	18.000

- ✓ La función `superanSalarioActividad01(empleado_02)` ¿continúa funcionando? Debería devolver



20.222.333	45	2	20.000
33.456.234	40	0	25.000
42.236.276	36	0	18.000

Actividad Nro. 03

- ✓ ¿Qué pasa si se modifica el orden de las columnas de la matriz?

Probar con el siguiente ejemplo
(denominar a la nueva matriz
empleado_03).

DNI	Salario	Edad	Cantidad de hijos
20.222.333	20.000	45	2
33.456.234	25.000	40	0
45.432.345	10.000	41	1
43.967.304	12.000	37	0
42.236.276	18.000	36	0

- ✓ La función `superanSalarioActividad01` ¿continúa funcionando? En caso de no funcionar implementar una nueva función denominada `superanSalarioActividad03`, que tome como entrada `empleado_03` y que devuelva a aquellos empleados que ganan un salario > \$15.000. El orden de las columnas debe ser el original:

1. DNI; 2. Edad; 3. Cantidad de hijo; 4. Salario

- ✓ `superanSalarioActividad03(empleado_03)` debería devolver

20.222.333	45	2	20.000
33.456.234	40	0	25.000
42.236.276	36	0	18.000

Actividad Nro. 03



- ✓ ¿Qué pasa si se modifica el orden de las columnas de la matriz?

Probar con el siguiente ejemplo
(denominar a la nueva matriz
empleado_03).

DNI	Salario	Edad	Cantidad de hijos
20.222.333	20.000	45	2
33.456.234	25.000	40	0
45.432.345	10.000	41	1
43.967.304	12.000	37	0
42.236.276	18.000	36	0

- ✓ La función `superanSalarioActividad01` ¿continúa funcionando? En caso de no funcionar implementar una nueva función denominada `superanSalarioActividad03`, que tome como entrada `empleado_03` y que devuelva a aquellos empleados que ganan un salario > \$15.000. El orden de las columnas debe ser el original:

1. DNI; 2. Edad; 3. Cantidad de hijo; 4. Salario

- ✓ `superanSalarioActividad03(empleado_03)` debería devolver

20.222.333	45	2	20.000
33.456.234	40	0	25.000
42.236.276	36	0	18.000

Actividad Nro. 04

- ✓ ¿Qué pasa si a la matriz de entrada se la implementa como lista de columnas (en vez de lista de filas)? Probar con el siguiente ejemplo (denominar a la nueva matriz empleado_04). La matriz resultante debe seguir implementada como lista de filas.

DNI	Salario	Edad	Cantidad de hijos
20.222.333	20.000	45	2
33.456.234	25.000	40	0
45.432.345	10.000	41	1
43.967.304	12.000	37	0
42.236.276	18.000	36	0

- ✓ ¿Alguna de las funciones anteriores (superanSalarioActividad0...) funciona? En caso de no funcionar implementar una nueva función denominada superanSalarioActividad04, que tome como entrada empleado_04 y que devuelva a aquellos empleados que ganan un salario > \$15.000. El orden de las columnas debe ser el original: 1. DNI; 2. Edad; 3. Cantidad de hijo; 4. Salario

- ✓ superanSalarioActividad04(empleado_04) debería devolver

20.222.333	45	2	20.000
33.456.234	40	0	25.000
42.236.276	36	0	18.000

Actividad Nro. 04



- ✓ ¿Qué pasa si a la matriz de entrada se la implementa como lista de columnas (en vez de lista de filas)? Probar con el siguiente ejemplo (denominar a la nueva matriz empleado_04). La matriz resultante debe seguir implementada como lista de filas.

DNI	Salario	Edad	Cantidad de hijos
20.222.333	20.000	45	2
33.456.234	25.000	40	0
45.432.345	10.000	41	1
43.967.304	12.000	37	0
42.236.276	18.000	36	0

- ✓ ¿Alguna de las funciones anteriores (superanSalarioActividad0...) funciona? En caso de no funcionar implementar una nueva función denominada superanSalarioActividad04, que tome como entrada empleado_04 y que devuelva a aquellos empleados que ganan un salario > \$15.000. El orden de las columnas debe ser el original: 1. DNI; 2. Edad; 3. Cantidad de hijo; 4. Salario

- ✓ superanSalarioActividad04(empleado_04) debería devolver

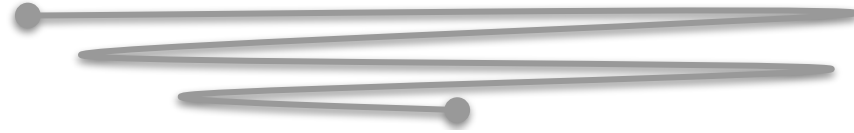
20.222.333	45	2	20.000
33.456.234	40	0	25.000
42.236.276	36	0	18.000

Actividad Nro. 05



- ✓ Responder las siguientes preguntas (escribir las respuestas a modo de comentario dentro del script)
1. ¿Cómo afectó a la programación de la función cuando cambiaron levemente la matriz de empleado?
 - a. En el caso en que le agregaron más filas
 - b. En el caso en que le alteraron el orden de las columnas
 2. ¿Y cuando a empleado le cambiaron la forma de representar las matrices (de lista de filas a lista de columnas)?
 3. ¿Cuál es la ventaja, desde el punto de vista del usuario de la función, disponer de ella y no escribir directamente el código de la consulta dentro de su programa?

Cierre



1. *Modificaciones en estructura de datos*

-> *Cambios en programación de las consultas*

Tareas Pendientes



Terminar de resolver ...

1. [individual] - guía de ejercicios de "Python"
2. [individual] - guía de ejercicios de "Modelo Relacional"
3. [individual] - guía de ejercicios de "Modelado de Datos"
4. [grupal] - TP arbolado